High speed multiplier using Nikhilam Sutra algorithm of Vedic mathematics

R Dhanupriya
dhanupriya.r@gmail.com
[1]PG Scholar, VLSI, Gokarajurangaraju Institute of Engg & Technology, Miyapur, Rangareddy.

G V Subbareddy
gvsreddy2005@gmail.com
[2]Associate Professor, Dept of ECE, Gokarajurangaraju Institute of Engg& Technology, Miyapur, Rangareddy.

Abstract: **In this paper, an exportable application-explicit guidance set elliptic curve cryptography processor dependent on repetitive marked digit portrayal is proposed. The processor utilizes broad pipelining strategies for Karatsuba– Ofman strategy to accomplish high throughput augmentation. The proposed design of this paper investigation the rationale size, region and power utilization utilizing Xilinx 13.2. The expansion for the undertaking is Vedic Sutra – Nikhilam technique.**

File Terms— *Index Terms*— Application-specific instruction-set processor (ASIP), elliptic curve cryptography (ECC), field-programmable gate array (FPGA), Karatsuba–Ofman multiplication, redundant signed digit (RSD).

INTRODUCTION

In prime field ECC processors, convey free number juggling is important to maintain a strategic distance from extensive information ways caused via convey proliferation. Excess plans, for example, convey spare number-crunching (CSA), repetitive marked digits (RSDs) , or buildup number frameworks (RNSs) , have been used in different structures. Convey rationale or installed advanced flag preparing (DSP) hinders inside field programmable door exhibits (FPGAs) are additionally used in a few plans to address the convey proliferation issue. It is important to manufacture a proficient expansion information way since it is an essential activity utilized in other secluded math tasks. Particular increase is a fundamental task in ECC.

Two principle methodologies might be utilized. The first is known as interleaved measured augmentation utilizing Montgomery's technique. Montgomery increase is broadly utilized in executions where self-assertive bends are wanted. Another methodology is known as increase then-decrease and is utilized in elliptic bends worked over limited fields of Merssene primes. Merssene primes are the extraordinary sort of primes which consider productive secluded decrease through arrangement of increments and subtractions. So as to enhance the duplication procedure, some ECC processors utilize the partition and vanquish approach of Karatsuba–Ofman augmentations, where others utilize implanted multipliers and DSP obstructs inside FPGA textures.

This paper proposes another RSD-based prime field ECC processor with rapid working recurrence. In this paper, we exhibit the execution of left-to-right scalar point increase calculation. The general processor engineering is of normal cross bar type with 256 digit wide information transports. The plan methodology and enhancement strategies are engaged toward proficient individual secluded math modules instead of the general engineering.

The staying of this paper is sorted out as pursues. Segment II gives foundation data on ECC frameworks. Segment III exhibits the general design of the proposed processor, the engineering of the particular number juggling unit (AU) is introduced. In Section IV, expansion of the task is talked about. At long last, Results and end is attracted Section V and Section VI.

I. RELATED WORK

Karatsuba– Ofman Multiplication:

The multifaceted nature of the normal duplication utilizing the textbook technique is O(n2). Karatsuba and Ofman proposed a philosophy to play out an increase with multifaceted nature O(n1.58)by

partitioning the operands of the augmentation into littler and break even with portions. Having two operands of length n to be duplicated, the Karatsuba–Ofman procedure recommends to part the two operands into high-(H)and low-(L) portions.

$$a_H = (a_{n-1}, \ldots, a_{\lceil n/2 \rceil}), \quad a_L = (a_{\lceil n/2 \rceil - 1}, \ldots, a_0)$$
$$b_H = (b_{n-1}, \ldots, b_{\lceil n/2 \rceil}), \quad b_L = (b_{\lceil n/2 \rceil - 1}, \ldots, b_0).$$

Consider β as the base for the operands, where β is 2 if there should arise an occurrence of numbers and β is x in the event of polynomials. At that point, the augmentation of the two operands is executed as pursues: considering

$$a = a_L + a_H \beta^{\lceil n/2 \rceil} \text{ and } b = b_L + b_H \beta^{\lceil n/2 \rceil}$$ then

$$C = AB = (a_L + a_H \beta^{\lceil n/2 \rceil})(b_L + b_H \beta^{\lceil n/2 \rceil}) = a_L b_L + (a_L b_H + a_H b_L)\beta^{\lceil n/2 \rceil} + a_H b_H \beta^n.$$

Henceforth, four half-sized augmentations are required, where Karatsuba approach reformulate (6) to

$$C = AB = (a_L + a_H \beta^{\lceil n/2 \rceil})(b_L + b_H \beta^{\lceil n/2 \rceil}) = a_L b_L + ((a_L + a_H)(b_L + b_H) - a_H b_H - a_L b_L)\beta^{\lceil n/2 \rceil} + a_H b_H \beta^n.$$

In this manner, just three half-sized augmentations are required. The first Karatsuba calculation is performed recursively, where the operands are sectioned into littler parts until the point when a sensible size is come to, and after that customary augmentations of the littler portions are performed recursively.

Excess Signed Digits:

The RSD portrayal, first presented by Avizienis [32], is a convey free math where whole numbers are spoken to by the distinction of two different numbers. A whole number X is spoken to by the distinction of its x+ and x− segments, where x+ is the positive part and x− is the negative segment. The idea of the RSD portrayal has the benefit of performing expansion and subtraction without the need of the two's supplement portrayal. Then again, an overhead is acquainted due with the excess in the number portrayal, since a whole number in RSD portrayal requires twofold word length contrasted and run of the mill two's supplement portrayal. In radix-2 adjusted RSD spoke to numbers, digits of such whole numbers are either 1, 0, or −1.

III.PROPOSED METHODOLOGY

The proposed P256 ECC processor comprises of an AU of 256 RSD digit wide, a limited state machine (FSM), memory, and two information transports. The processor can be designed in the pre-amalgamation stage to help the P192 or P224 NIST prescribed prime bends [36]. Fig. 1 demonstrates the general processor design. Two sub control units are connected to the fundamental control unit as extra squares. These two sub control units fill in as FSMs for point expansion and point multiplying, individually. Diverse arrange frameworks are effectively bolstered by including relating sub control hinders that work as indicated by the recipes of the facilitate framework.

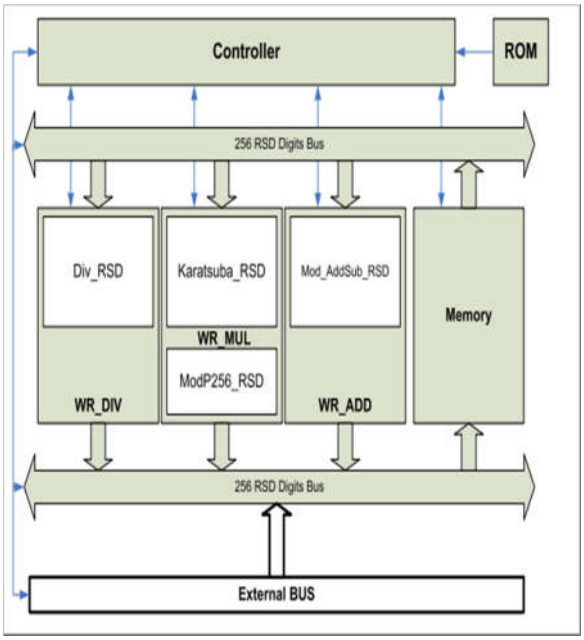


Fig. 1. Overall processor architecture.

Number juggling UNIT.

Particular Addition and Subtraction Addition is utilized in the gathering procedure amid the increase, and also, in the parallel GCD secluded divider calculation. In the proposed usage, radix-2 RSD portrayal framework as convey free portrayal is

utilized. In RSD with radix-2, digits are spoken to by 0, 1, and −1, where digit 0 is coded with 00, digit 1 is coded with 10, and digit −1 is coded with 01. In Fig. 2, a RSD snake is introduced that is worked from summed up full adders.

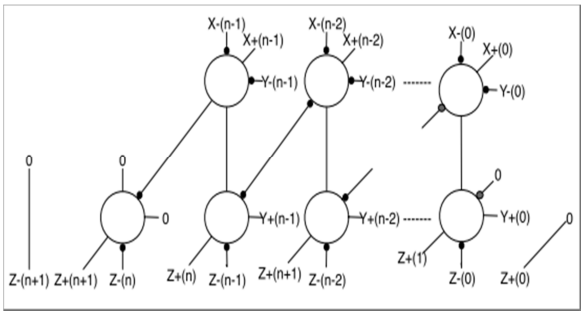


Fig. 2. RSD adder.

Secluded Multiplication

Karatsuba's multiplier recursive nature is viewed as a noteworthy downside when actualized in equipment. Equipment multifaceted nature increments exponentially with the measure of the operands to be duplicated. To conquer this downside, Karatsuba technique is connected at two dimensions. A recursive Karatsuba hinder that works profundity insightful, and an iterative Karatsuba that works widthwise.

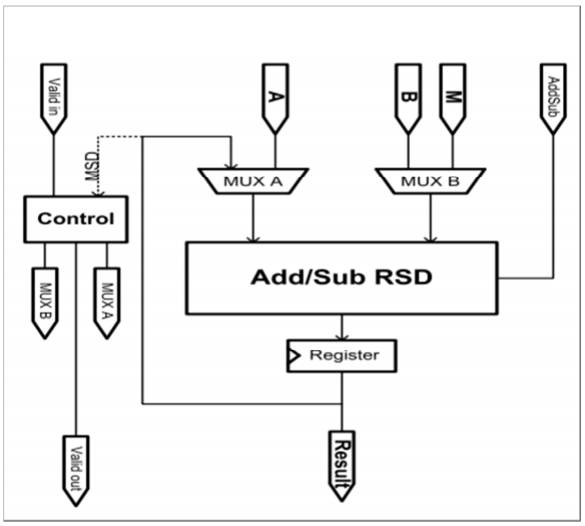


Fig. 3. Modular addition subtraction block diagram.

The square graph of the recursive Karatsuba multiplier is appeared in Fig. 4, where information conditions are obviously taken note. As appeared in Fig. 4, Karatsuba technique requires playing out a subtraction at each dimension, which is leeway of the proposed usage since subtraction is performed with no additional expense in RSD portrayal. The square chart of the recursive Karatsuba module is worked from three half-sized recursive Karatsuba squares and some RSD adders/subtracters. There is one 1-digit RSD multiplier that is utilized to increase the convey digits from the center expansion. As indicated by Fig. 4, the basic datapath of the recursive Karatsuba is partitioned into two ways. The main way experiences the center half-sized recursive Karatsuba square, and alternate experiences the cross result of the center expansion with multiplexers and a few adders.

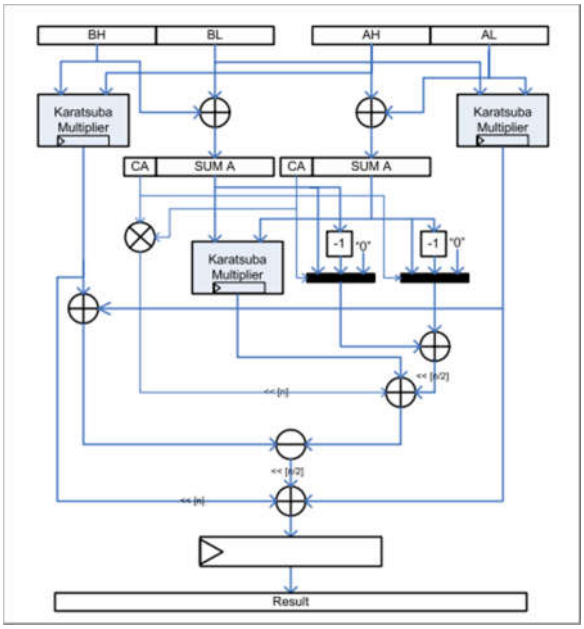


Fig. 4. Karatsuba recursive block

NIST Reduction: Generalized Mersenne primes [19] are the extraordinary kind prime numbers that permit quick measured decrease. Normal division is supplanted by couple of increases and subtractions. Such primes are spoken to as p = f (t), where t is an intensity of 2. The modulus of the P256 bend is Merssene prime

$p = 2^{256}-2^{224}+2^{192}+2^{96}-1$.

Because of the excess idea of the RSD portrayal, the increase procedure may create results that are spoken to by in excess of 512 digits and these outcomes are still in the range $-p2 < A < p2$. These a couple of additional digits are outside the scope of the NIST decrease process. Thus, we determined new equations to incorporate these additional digits in the

decrease procedure. The new decrease process has one additional 256-digit term, D5, alongside some change of the recently existed terms. This term is included restrictively, regardless of whether the additional digit is set or not. Along these lines, two augmentations are the aggregate overhead required to deal with the additional digits caused utilizing the RSD portrayal. The changed decrease recipe is B = T + 2S1 + 2S2 + S3 + S4 − D1 − D2 − D3 − D4 − D5 mod p, where A16 speaks to the additional digits created by RSD Karatsuba multiplier.

$$
\begin{aligned}
T &= (A_7 \| A_6 \| A_5 \| A_4 \| A_3 \| A_2 \| A_1 \| A_0) \\
S_1 &= (A_{15} \| A_{14} \| A_{13} \| A_{12} \| A_{11} \| 0 \| 0 \| 0) \\
S_2 &= (\mathbf{2 * A_{16}} \| A_{15} \| A_{14} \| A_{13} \| A_{12} \| 0 \| 0 \| \mathbf{A_{16}}) \\
S_3 &= (A_{15} \| A_{14} \| 0 \| 0 \| \mathbf{-2 * A_{16}} \| A_{10} \| A_9 \| A_8) \\
S_4 &= (A_8 \| A_{13} \| A_{15} \| A_{14} \| A_{13} \| A_{11} \| A_{10} \| A_9) \\
D_1 &= (A_{10} \| A_8 \| 0 \| 0 \| \mathbf{2 * A_{16}} \| A_{13} \| A_{12} \| A_{11}) \\
D_2 &= (A_{11} \| A_9 \| 0 \| \mathbf{A_{16}} \| A_{15} \| A_{14} \| A_{13} \| A_{12}) \\
D_3 &= (A_{12} \| \mathbf{2 * A_{16}} \| A_{10} \| A_9 \| A_8 \| A_{15} \| A_{14} \| A_{13}) \\
D_4 &= (A_{13} \| 0 \| A_{11} \| A_{10} \| A_9 \| \mathbf{A_{16}} \| A_{15} \| A_{14}) \\
\mathbf{D_5} &= \mathbf{(-A_{16} \| 0 \| 0 \| 0 \| 0 \| 0 \| 0 \| -A_{16})}.
\end{aligned}
$$

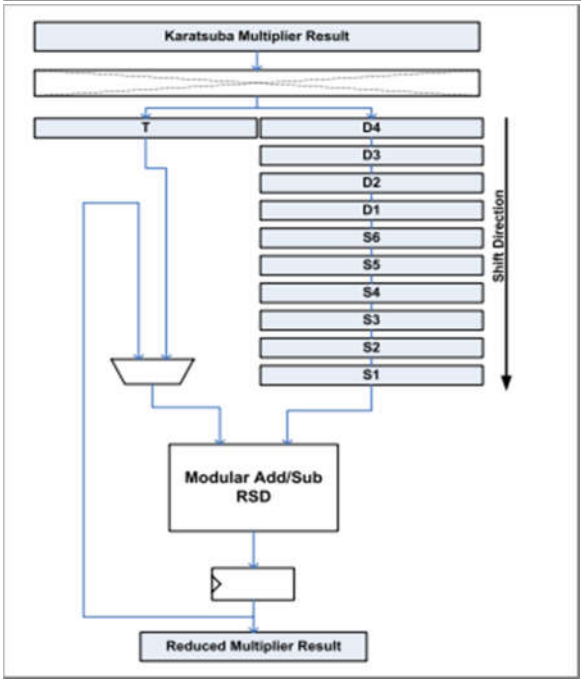


Fig. 5. Mod P256 reduction block.

So as to oblige the additional digit delivered by the RSD Karatsuba multiplier, NIST decrease is reformulated. The resultant decrease plot comprises of three additional options. Be that as it may, through reformulation and consolidating the first terms with the extra terms, the decrease conspire is upgraded. As needs be, the secluded multiplier is worked with a Karatsuba multiplier, particular RSD snake, and a few registers to hold the 256-digit terms. Fig. 5 demonstrates the square chart of the Mod P256 RSD multiplier. A controller is utilized to control the stream of the terms to the secluded snake and every step of the way, the aftereffect of the particular expansion is aggregated and encouraged back to the viper. The cross-bar in Fig. 5 demonstrates the wiring of the 32-digit words to their separate areas inside the all-inclusive NIST decrease registers.

High-Radix Modular Division

Twofold GCD calculation is a productive method for performing particular division since it depends on expansion, subtraction, and moving tasks. The multifaceted nature of the division task originates from the way that the running time of the calculation is conflicting and is input subordinate.

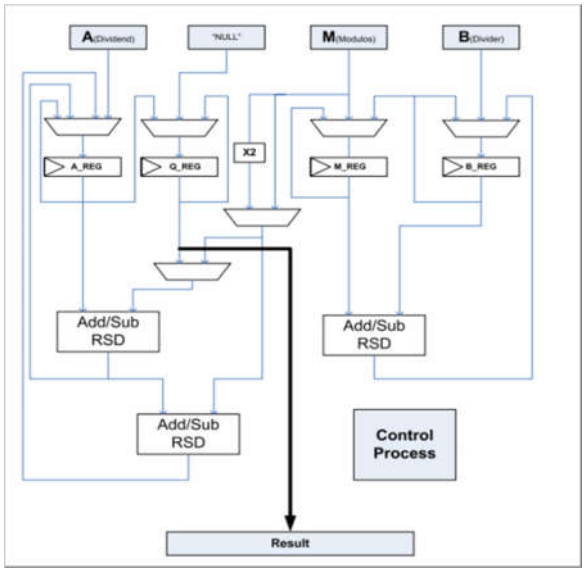


Fig. 6. Modular divider block

IV. Vedic Sutra – Nikhilam method

Nikhilam Sutra is one of the 16 sutras of Vedic science. It tends to be utilized to change over extensive digits augmentation to little digits duplication with the assistance of couple of additional include, subtract and move activities. Now and again two-digit increase can be performed utilizing just 1 one-digit duplication rather than 3 one-digit augmentation as required by Karatsuba calculation.

Assume we need to perform same duplication 95*96 utilizing this strategy. We can utilize the Nikhilam sutra as pursues:

1. Register A = 100-95; Subtract the multiplicand from closest base

2. Register B = 100-96; Subtract the multiplier from a similar base

3. Register C = B * A = 5 * 4 = 20

4. Register D = 95 - 4 = 96 □ 5 = 91

5. Result 100*D + C = 9120

In the Fig. 7, we can see that there is just a single duplication task included.

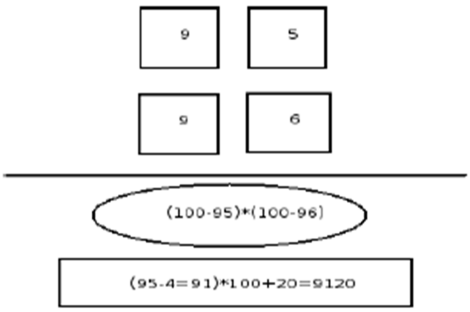


Fig. 7. Multiplication of integers (95 _ 96) using Nikhilam method

Above multiplication is also shown in Table 1.

TABLE I
MULTIPLICATION OF 95 * 96

	Integer	Base Difference
Multiplicand	95	(100-95)=5
Multiplier	96	(100-96)=4
	(95-4)=91	(5*4)=20
Result	9120	

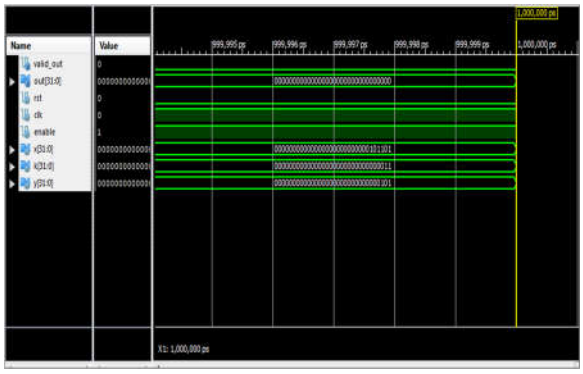
In this increase we have utilized 1 augmentation, 1 expansion, 3 subtraction and 1 move activity. This specific duplication is more proficient than both standard augmentation and Karatsuba technique. Assume multiplicand is m = x*a and multiplier is n = x * b where x is closest base. We have:

$$m * n = (x - a) * (x - b) = x(x - a - b) + ab$$

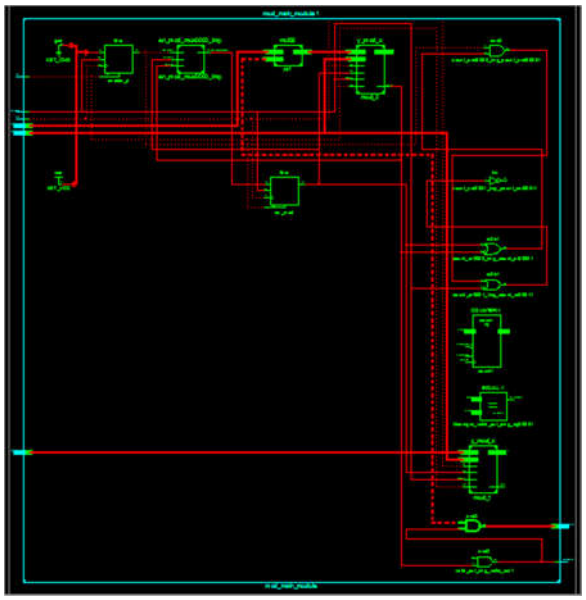
V.EXPERIMENTAL RESULTS

Results of proposed method

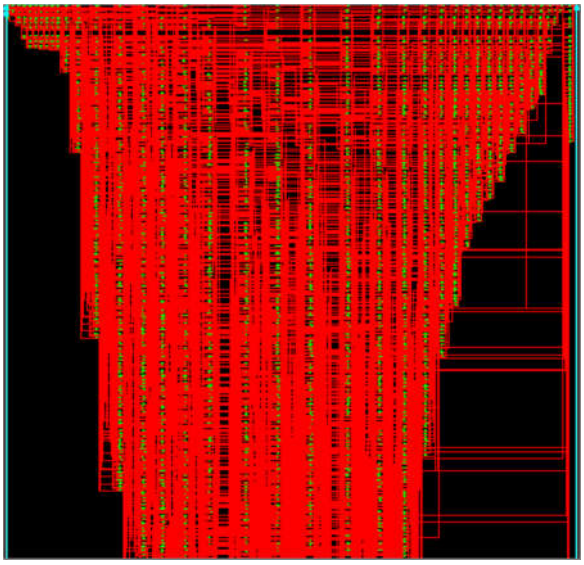
Simulation.



RTL Schematic.



Technology Schematic.



Design Summary.

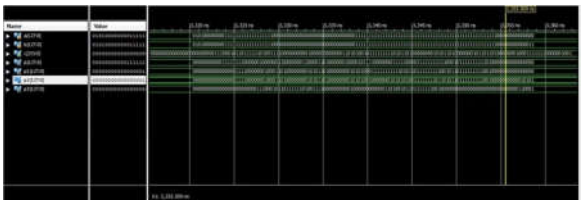
Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	930	4656	19%
Number of Slice Flip Flops	177	9312	1%
Number of 4 input LUTs	1741	9312	18%
Number of bonded IOBs	132	232	56%
Number of GCLKs	1	24	4%

Timing Report.

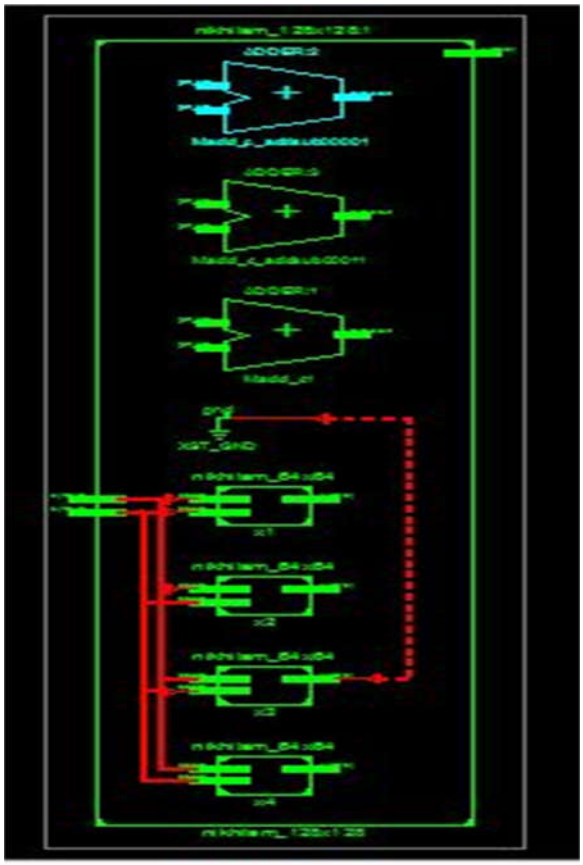
```
Timing constraint: Default OFFSET OUT AFTER for Clock 'clk'
  Total number of paths / destination ports: 2177 / 33
-------------------------------------------------------------------------
Offset:              8.440ns (Levels of Logic = 19)
  Source:            mod_2/z_0 (FF)
  Destination:       valid_out (PAD)
  Source Clock:      clk rising
```

Extension.

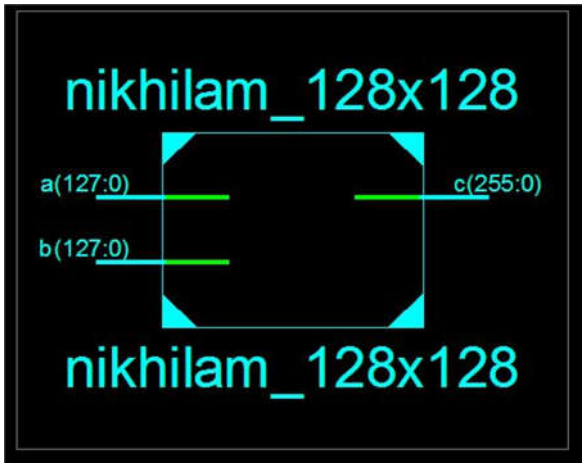
Simulation.



RTL Schematic.



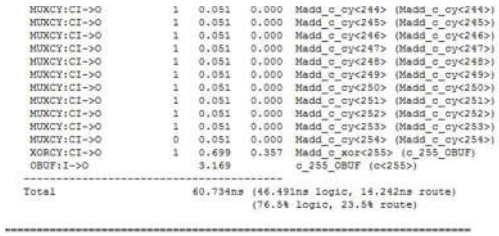
Technology Schematic.



Design Summary.

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	3512	4656	75%
Number of 4 input LUTs	6619	9312	71%
Number of bonded IOBs	448	232	193%

Timing Report.



VI.CONCLUSION

In this paper, a NIST 256 prime field ECC processor execution in FPGA has been introduced. A RSD as a convey free portrayal is used which brought about short datapaths and expanded greatest recurrence. We presented upgraded pipelining methods inside Karatsuba multiplier to accomplish high throughput execution by a completely LUT-based FPGA usage.. Moreover, an effective measured expansion/subtraction is presented dependent on checking the LSD of the operands as it were. A control unit with extra like design is proposed as a reconfigurability highlight to help diverse point increase calculations and facilitate frameworks. The primary focal points of our processor incorporate the exportability to other FPGA and ASIC advancements and expandability to help diverse arrange frameworks and point increase calculations.

REFERENCES

[1] N. Koblitz, “Elliptic curve cryptosystems,” Math. Comput., vol. 48, no. 177, pp. 203–209, Jan. 1987.

[2] W. Stallings, Cryptography and Network Security: Principles and Practice, 5th ed. Englewood Cliffs, NJ, USA: Prentice-Hall, Jan. 2010.

[3] C. Rebeiro, S. S. Roy, and D. Mukhopadhyay, “Pushing the limits of high-speed GF(2m) elliptic curve scalar multiplication on FPGAs,” in Proc. Cryptograph. Hardw. Embedded Syst. (CHES), vol. 7428. Jan. 2012, pp. 494–511.

[4] Y. Wang and R. Li, “A unified architecture for supporting operations of AES and ECC,” in Proc. 4th Int. Symp. Parallel Archit., Algorithms Programm. (PAAP), Dec. 2011, pp. 185–189. [5] S. Mane, L. Judge, and P. Schaumont, “An integrated prime-field ECDLP hardware accelerator with high-performance modular arithmetic units,” in Proc. Int. Conf. Reconfigurable Comput. FPGAs, Nov./Dec. 2011, pp. 198–203.

[6] M. Esmaeildoust, D. Schinianakis, H. Javashi, T. Stouraitis, and K. Navi, “Efficient RNS implementation of elliptic curve point multiplication over GF(p),” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 21, no. 8, pp. 1545–1549, Aug. 2012.

[7] D. M. Schinianakis, A. P. Fournaris, H. E. Michail, A. P. Kakarountas, and T. Stouraitis, “An RNS implementation of an Fp elliptic curve point multiplier,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 56, no. 6, pp. 1202–1213, Jun. 2009.

[8] J.-W. Lee, S.-C. Chung, H.-C. Chang, and C.-Y. Lee, “Efficient poweranalysis-resistant dual-field elliptic curve cryptographic processor using heterogeneous dual-processing-element architecture,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 22, no. 1, pp. 49–61, Feb. 2013.

[9] J.-Y. Lai and C.-T. Huang, “Energy-adaptive dual-field processor for high-performance elliptic curve cryptographic applications,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 8, pp. 1512–1517, Aug. 2011.

[10] S.-C. Chung, J.-W. Lee, H.-C. Chang, and C.-Y. Lee, “A highperformance elliptic curve cryptographic processor over GF(p) with SPA resistance,” in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2012, pp. 1456–1459.

[11] J.-Y. Lai and C.-T. Huang, “Elixir: High-throughput cost-effective dualfield processors and the design framework for elliptic curve cryptography,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 16, no. 11, pp. 1567–1580, Nov. 2008.

[12] D. Karakoyunlu, F. K. Gurkaynak, B. Sunar, and Y. Leblebici, “Efficient and side-channel-aware implementations of elliptic curve cryptosystems over prime fields,” IET Inf. Secur., vol. 4, no. 1, pp. 30–43, Mar. 2010.

[13] D. Schinianakis and T. Stouraitis, “Multifunction residue architectures for cryptography,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 61, no. 4, pp. 1156–1169, Apr. 2014.

[14] J. Vliegen et al., “A compact FPGA-based architecture for elliptic curve cryptography over prime fields,” in Proc. 21st IEEE Int. Conf. Appl.-Specific Syst. Archit. Process. (ASAP), Jul. 2010, pp. 313–316.

[15] T. Güneysu and C. Paar, “Ultra high performance ECC over NIST primes on commercial FPGAs,” in Proc. 10th Int. Workshop Cryptograph.

Hardw. Embedded Syst. (CHES), 2008, pp. 62–78.
[16] P. L. Montgomery, "Modular multiplication without trial division," Math. Comput., vol. 44, no. 170, pp. 519–521, Apr. 1985.
[17] K. Sakiyama, N. Mentens, L. Batina, B. Preneel, and I. Verbauwhede, "Reconfigurable modular arithmetic logic unit for high-performance public-key cryptosystems," in Proc. 2nd Int. Workshop Reconfigurable Comput., Archit. Appl., vol. 3985. 2006, pp. 347–357.
[18] A. Byrne, E. Popovici, and W. P. Marnane, "Versatile processor for GF(pm) arithmetic for use in cryptographic applications," IET Comput. Digit. Tech., vol. 2, no. 4, pp. 253–264, Jul. 2008.
[19] J. Solinas, "Generalized Mersanne number," Univ. Waterloo, Waterloo, ON, Canada, Tech. Rep. CORR 99-39, 1999. [20] B. Ansari and M. A. Hasan, "High-performance architecture of elliptic curve scalar multiplication," IEEE Trans. Comput., vol. 57, no. 11, pp. 1443–1453, Nov. 2008.
[21] N. Smyth, M. McLoone, and J. V. McCanny, "An adaptable and scalable asymmetric cryptographic processor," in Proc. Int. Conf. Appl.-Specific Syst., Archit. Processors (ASAP), Sep. 2006, pp. 341–346.
[22] C. J. McIvor, M. McLoone, and J. V. McCanny, "Hardware elliptic curve cryptographic processor over GF(p)," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 53, no. 9, pp. 1946–1957, Sep. 2006.
[23] K. Ananyi, H. Alrimeih, and D. Rakhmatov, "Flexible hardware processor for elliptic curve cryptography over NIST prime fields," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 17, no. 8, pp. 1099–1112, Aug. 2009.
[24] M. Hamilton and W. P. Marnane, "FPGA implementation of an elliptic curve processor using the GLV method," in Proc. Int. Conf. Reconfigurable Comput. FPGAs (ReConFig), Dec. 2009, pp. 249–254.
[25] F. Crowe, A. Daly, and W. Marnane, "A scalable dual mode arithmetic unit for public key cryptosystems," in Proc. Int. Conf. Inf. Technol., Coding Comput. (ITCC), vol. 1. Apr. 2005, pp. 568–573.
[26] N. Takagi, "A VLSI algorithm for modular division based on the binary GCD algorithm," IEICE Trans. Fundam. Electron., Commun., Comput. Sci., vol. E81-A, no. 5, pp. 724–728, May 1998.